

**PATENT**  
**IBM Docket No. UK9-98-0128**

**REMARKS**

**Status:**

Claims 1-23 stand rejected under 35 U.S.C. §112 (e) as being anticipated by the teaching of US Pat. No. 6,119,,195 to Goodwin et al.

Claims 1-23 are presented for reconsideration as is explained in the discussion that follows.

**Discussion:**

Considering firstly the Goodwin teaching, the focus appears to be consistently at the object development level. (In the following, quoted material is highlighted in italics and emphasis is added by adding bolding) This is consistent with Applicant's Background of the Invention (page 2 lines 8-10); i.e. "*It is well-known in the art to have objects in classes with common run-time characteristics determined at the class level.*"

Again considering the Goodwin teaching: the "Background of the Invention" second sentence states:

*"Even more particularly, the present invention relates to automatically generated source code objects within extensible object frameworks and links to enterprise resources".*

The "Summary of the Invention" section begins:

*"The present invention advantageously provides an approach for automatically generating source code, and specifically for designing and authoring source code within a complex business framework, and generating business objects with all implemented behaviors within a composed object service framework".*

**PATENT**  
**IBM Docket No. UK9-98-0128**

At col. 3, starting at line 4, Goodwin states "*In one embodiment, the invention can be characterized as a method for generating source code objects.* And, at line 13, same page, "*In another embodiment, the invention can be characterized as a system for generating source code objects.*"

In the Detailed Description (col. 4, lines 1-13) Goodwin explains:

*"An "object class" is a set of data (attributes) and functional capabilities (routines) encapsulated into a single logical entity. For example, a "drug treatment" class may be characterized by a "dosage" attribute and an "administer-medication" routine.*

*An "object instance" is an embodiment (instantiation) of an object class. Instances are differentiated from one another by their attribute values, but not their routines (capabilities). For example, Jane Smith may be a first patient-object instance and Jon Doe may be a second patient-object instance. The term "object" is often used by itself to refer loosely to either an object class or an object instance, the difference being understood in context"*

Goodwin recognizes the proper use of object and, in the context of the teaching appears not to be referring to instances having capabilities.

At Goodwin, col 7, lines 22-28 it is stated:

*"Large complex systems are composed of hundreds, if not thousands of business objects, and, various service specifications, such as CORBA common object service specifications (i.e., security, transaction, life-cycle, proxy, etc.), and require methods and behaviors to be implemented on each business object.*

**PATENT**  
**IBM Docket No. UK9-98-0128**

At Goodwin, col 7, lines 50-62 it is indicated:

***"An important feature of the present embodiment is an ability to help the developer, at development-time, to rapidly generate business objects that are composed of behaviors from various object management services (persistence state, proxy object, life cycle, externalization, event, etc.) using an open framework. The present embodiment also allows developers to define unique (i.e., custom) services and unique behaviors that can be integrated into an object framework. At run-time, in the present embodiment, object management services utilize the objects that were generated using the code generator 210 to retrieve user data from distributed data sources as objects."***

It appears clear that Goodwin is teaching an improved way to develop objects at development time and the teaching makes clear the distinction between an object and an object instance. The references to "lifecycle" refer to objects not instances.

Applicants, on the other hand, have recognized that instances may advantageously be used for multiple transactions if they are given respective behaviors such as lifecycles. For example, in banking, a debit-credit pair activity is a common transaction that is processed. When performed at an ATM the activity may have to be processed immediately; whereas, a counter transaction may be accumulated for bulk processing overnight. Processing time is saved if a new instance need not be created and destroyed for each transaction. Applicant's multi-message instances (page 12 of Applicant's specification) with lifecycle descriptors would allow multiple debit-credit pairs to be accumulated for processing at a later time.

**PATENT**  
**IBM Docket No. UK9-98-0128**

As Applicant's indicate at Specification page 4, line 13-23:

*"In the Enterprise Java Bean environment, the client creates beans (instances) when they are needed for a particular processing action, and destroys them when they are no longer needed for the particular processing action. This can be wasteful in its consumption of resources: where many clients are initiating and terminating processing in a busy environment, the system can be consuming considerable processing resource in the creation and destruction of these instances..."*

*Accordingly, in a first aspect, the present invention provides apparatus for processing business data processing activities, said apparatus comprising: an activity instance; and a descriptor for said activity instance associated uniquely with said activity instance; herein said descriptor includes parameters for determining lifecycle behaviors of said activity instance.*

*Said descriptor is preferably programmable to modify said lifecycle behaviors according to a use made of said activity instance."*

This lifestyle descriptor uniquely associated for determining lifecycle behavior is emphasized in all of the claims (see e.g. independent claims 1, 10 and 19). And, the ability to vary lifecycle according to use is emphasized in claims 2, 11 and 20 and those claims dependent upon them. Accordingly, it is believed that the claims clearly identify inventive subject matter not taught or suggested in the prior art.

**PATENT**  
**IBM Docket No. UK9-98-0128**

Consistant with the above it is believed this Application has been placed in condition for allowance and early notice to that effect is earnestly solicited.

Respectfully Submitted,



George Z. Grosser

Reg. No. 25,629

c/o IBM Corp.  
Dept. T81/Bldg. 503 PO Box 12195  
Research Triangle Park, NC 27709

(919)968-7847  
Fax 919-254-4330  
EMAIL: gegch@prodigy.net